LLM-driven Interactive document classification through Keyword Feedback

Boan Yu, Mei Wang, Dehua Chen, Qiao Pan and Yunhua Wen (✉)

Donghua Unibersity, Shanghai 201620, China
2212503@mail.dhu.edu.cn, {wangmei, chendehua, panqiao, yhwen}@dhu.edu.cn

Abstract. Document classification offers a concise comprehension of document content, which is crucial for document organization and management in real application. However, practical scenarios pose challenges due to limited annotated data and dynamic changes in document categories. In this paper, we propose an LLM-driven interactive document classification framework based on keyword feedback, which operates with minimal input—just the documents to be classified. We achieve this by first introducing an unsupervised learning based document classification framework. Then a keyword interaction process is designed to iteratively enhance the classifier's performance. The representative keyword explanations is generated in each iteration, which offer the most significant features or characteristics within each category. Crucially, an LLM feedback module is designed for interaction which offers category description and keyword feedback, facilitating seamless cooperation to enhance classification performance. Experimental results on benchmark datasets demonstrated that our framework significantly improves classifier accuracy when compared to methods lacking feedback with few feedback iterations.

Keywords: Interactive document classification, keyword feedback, LLM.

1 Introduction

With the advancement of information technology, governments and businesses are generating a substantial amount of text at every moment. Document classification plays a crucial role in providing a concise understanding of document content, enabling effective organization and management. In recent years, numerous deep learning approaches [1] applied to document classification task have yielded excellent outcomes. However, within practical application scenarios, the dynamic changes in document categories and the substantial lack of annotated data have presented significant challenges to these techniques. How to accurately classify documents under these challenges is an urgent real-world issue.

In the context of training classifiers with unlabeled data, a variety of techniques have emerged, including unsupervised learning based methods [2,3], weakly supervised learning based methods [4], and interactive learning based methods [5,6]. A common approach in unsupervised or weakly supervised classification is to construct a pseudo training dataset through clustering or with the help of the supervision information like external knowledge base [7] or seed words [8] before applying supervised classification algorithm. Traditional supervised classification techniques are then employed for categorization. However, these methods also have some limitations. Users may not have seen all documents initially, making initial seed information somewhat arbitrary and less accurate. Consequently, the generate pseudo-labels often contain considerable noise, resulting in inaccurate outcomes in supervised classification. Clearly, in both cases, the classification accuracy, which is initially low, could potentially be improved through further interaction with users. When considering interactive methods, most approaches involve presenting users with samples that have a low output certainty from the guiding classifier for feedback, thereby enhancing the accuracy of labeled training samples. However, providing feedback labels for documents requires users to review the entire document, which can be labor-intensive and result in insufficient feedback.

The emergence of text-generating large language models (LLMs) [9,10,11,12] presents a potential solution, namely, automated feedback generation. LLMs are advanced AI systems that have been trained on vast amounts of textual data using the transformer architecture and the attention mechanism. This enables them to learn statistical relationships between words, phrases, context and topic information. However, documents from entities such as governments and businesses often contain sensitive information, making it essential to adhere to strict privacy and security standards. Directly interacting with documents may lead to privacy and security risks. Training a local LLM also requires a significant amount of training data. Therefore, this paper proposes to explore the potential of LLM-driven interaction on keyword feedback and study its impact on classification performance. In fact, representative keywords play a pivotal role in identifying the most significant features or characteristics within each category, aiding in understanding the topic or content in each category. Our objective is to seamlessly integrate the proposed interaction mechanism with the document classification framework, achieving accurate results with minimal or no human intervention.

To accomplish this, we propose an interactive document classification framework driven by LLM, which relies on keyword feedback. Initially, we utilize unsupervised clustering and refinement process to derive pseudo labels for documents. Subsequently, a BERT-based classifier is trained on these pseudo labels and documents, following a self-supervised learning framework. Once the self-supervised learning is finished, an LLM driven interaction is provided. During the interactive round, LLM generates category descriptions and provides feedback on the keyword list to filter out keywords that do not belong to the class, which improve the quality of the training data and the subsequent classifier training.

The contributions of this paper are as follows:

1) We propose an LLM-driven interactive document classification framework that delivers interpretable keywords for feedback, facilitating more effective feedback and model adjustments.
2) We design an LLM interactive interface to offer two types of feedback: distinguishing mixed categories in clustering as well as offering feedback on the

keyword list to filter out irrelevant keywords, helping improve the training set quality.

3) Experimental results on benchmark datasets demonstrated that our framework significantly improves classifier accuracy when compared to methods lacking feedback with few feedback iterations.

This paper is organized as follows. Section 2 describes the proposed framework in detail. Section 3 provides the experimental results. Finally, future work is discussed in Section 4.

2 Our Work

In this section, we first introduce the pipeline of the proposed method, then we provide the details of each stage.

2.1 Pipeline

Given the unlabeled document set $D = \{D_1, D_2, \ldots, D_n\}$ where n denotes the number of documents. Our goal is to build a document classifier to assign the label or category descriptions to the document in D. To this end, we propose an LLM-driven interactive Document Classification (LLM-DC) framework based on keyword feedback. As illustrated by Fig. 1, our LLM-DC is composed of three seamless components: (a) Initial Label Generation; (b) Self-supervised classification and (c) LLM-driven Feedback.

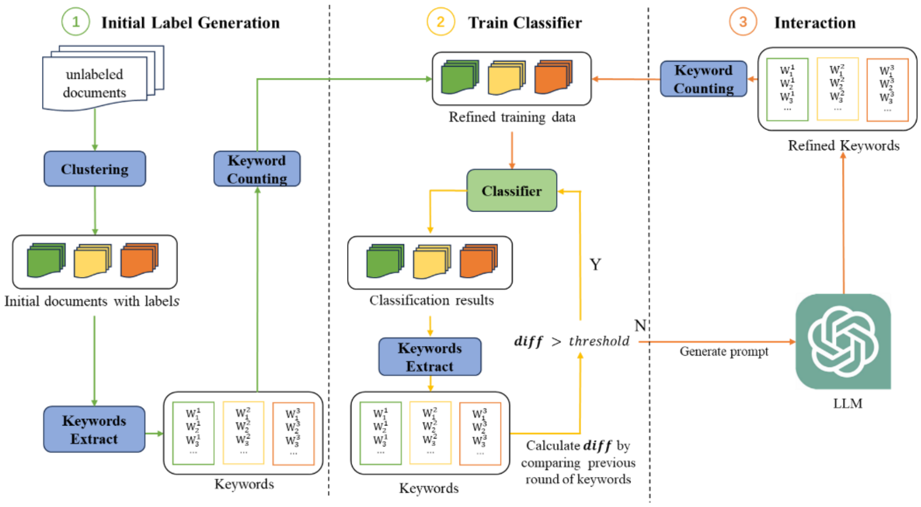

Fig. 1. The Framework Pipeline: Initial Label Generation (green line), Self-supervised Classification (yellow line), and LLM-driven Feedback (orange line)

For initial label generation, we utilize unsupervised clustering followed by a refinement process to generate pseudo-labels for the documents. In this way, the initial training set PD^0 is obtained. in classifier training, we employ a self-supervised learning framework, where its performance continuously improves through iterative training based on the classifier's own generated results. Considering the initial label generation, we rely solely on unsupervised methods, inevitably leading to instances where documents from different classes are assigned to the same cluster, resulting in initial label errors. Correcting such errors proves challenging within the self-supervised classification training paradigm. Consequently, LLM-Driven interactive feedback is designed to deal with this problem. The entire interaction is divided into two categories. During the initial interactive round, LLM generates category descriptions and assists in reconstructing categories, establishing the foundation for further enhancements. In the subsequent round, LLM provides keyword feedback. Utilizing this feedback, the document labels are reassigned and initiate another round of training for the self-learning classifier, thereby iteratively improving the performance.

2.2 Initial Label Generation

Considering that the number of categories cannot be precisely determined beforehand, we need to adopt a clustering method that does not require specifying the number of categories to generate initial labels. To fulfill this requirement, we employ DocSCAN [13], which operates as a fully unsupervised text classification approach leveraging semantic clustering through the nearest neighbours algorithm. For each document, we extract embedding vectors from a large pre-trained language model SBERT. It has been demonstrated that, in the embedding space, neighboring documents frequently exhibit similar class labels. Consequently, the model can autonomously learn document topic classification through the relationships within the neighborhood.

Following the initial clustering step, the document set D is partitioned into N clusters. We then proceed to extract representative keywords for each cluster, employing an enhanced TF-IDF method [14]. The words with the highest TF-IDF scores are selected as the keywords for the respective cluster. These keywords serve as representatives of the predominant information within each cluster and are denoted as $K = \{K_1, K_2, \ldots, K_N\}$, where $K_i = \{w^i_1, w^i_2, \ldots, w^i_m\}$, with each w_m denotes a keyword. Then a keyword counting strategy is employed using K^0 to create our training set PD^0, which calculate the occurrence of each keyword within the document and then reassigns the document to the category with the highest cumulative occurrence. The resulting label l_i of a given document D_i are generated using this keyword counting process as outlined below:

$$l_i = \underset{\theta}{\operatorname{argmax}} \left\{ \sum_j tf\left(w_j, D_i | \forall\left(w_j \in K_\theta\right)\right) \right\} \tag{1}$$

The recent study [14] proposed to improve the accuracy of pseudo-label generation by training Graph Neural Networks (GNN) to model the correlations between keywords. However, their approach involved fine-tuning the GNN based on the results of keyword counting. Therefore, this paper chooses to directly employ the simple yet effective keyword counting method.

2.3 Self-supervised Classification

Based on the obtained initial training data, we train a document classifier based on BERT [15]. The training process of the classifier follows a self-supervised learning framework, where its performance continuously improves through iterative training based on its own generated results. In the training iteration, the classifier is fine-tuned based on PD^{cur} (The *cur* start with 0). After training, the classifier is applied to classify the documents in D, resulting in PD^{cur+1}. Subsequently, keywords K^{cur+1} are extracted from PD^{cur+1} using the same method as described in Section 2.2. The iterative training process continues until the difference between K^{cur+1} and K^{cur} falls below a predefined threshold. The calculation of this difference, labeled as $diff_K$, is carried out as follows:

$$diff_K = \frac{NUM(K^{pre} \cup K^{cur} - K^{pre} \cap K^{cur})}{NUM(K^{pre} \cup K^{cur})} \qquad (2)$$

2.4 LLM-driven Feedback

Considering the initial label generation, we rely solely on unsupervised methods, which inevitably introduce noise in initial training set. Such noise will significantly degrade the classification performance of the classifier in Section 2.3. To deal with this problem, an LLM-driven interactive feedback component is designed. Let K represent the list of representative keyword sets for each class obtained after self-supervised training. These keywords play a crucial role in identifying the most significant features or characteristics within each category. Different categories of keyword sets, such as K_i and K_j, should exhibit a certain level of distinctiveness. Additionally, errors caused by mis-clustering are reflected in the keyword set. Therefore, we aim for the designed LLM interface to offer two types of feedback: distinguishing mixed categories in clustering and generating category descriptions, as well as offering feedback on the keyword list to filter out irrelevant keywords. This process enhances the accuracy of the keyword set and further improves the quality of the training data.

The interaction process is illustrated in Algorithm 1. The algorithm takes as input the keyword set K obtained at the end of the previous classifier training, along with the current number of categories. During the initial interaction, it follows the subsequent steps (step 2 - step 7): Firstly, prompts are generated using keyword set K_i for category i and queried to LLM. LLM are tasked with generating topic descriptions based on the semantics of these keywords and splitting them into different topics. The designed prompt template is shown in Table 1. Next, all generated topics will be filtered and merged. To enhance the accuracy of merging, human feedback can be incorporated at this stage. Following this, keywords that likely belong to the same category are consolidated, and the resulting keyword set list *K_interact* and new category descriptions *Topic_name_list* are returned. This process aims to rectify category errors and refine the categorization structure. After the first round of interaction finishes, the *K_interact* returned by LLM is employed in Eqn. 1 for another round of keyword counting. The resulting training set is then regenerated and employed for classifier training. Upon reaching the conditions defined in Eqn. 2, the LLM interaction starts again. At this point, topic regeneration no longer occurs. The interaction feedback at this stage relies on LLM to filter keywords according to the topic descriptions. As illustrated in Algorithm 1 (step 9 - step 13), We first generate new prompts. LLM will provide feedback on the keyword list to filter out keywords that do not belong to the class. The prompt template is shown in Table 2. After the interaction finishes, a new round of dataset generation and classifier training begins, continuing until the performance difference between the two consecutive rounds falls below a specified threshold.

Algorithm 1 LLM-driven Interaction

```
Input: $K$: Keywords obtained from last training; $N$ : Current number of categories
Output: $K\_interact$: Keywords after feedback;
1:  if First Interaction then
2:      for $i$ from 1 to $N$ do
3:          $prompt = generate\_prompt\_1(K_i)$
4:          $Topic\_name\_list, Keywords\_list$ = LLM($prompt$)
5:      end for
6:      $K\_interact = combine\_Topic(Topic\_name\_list, Keywords\_list)$
7:      return $K\_interact$
8:  else
9:      for $i$ from 1 to $\underline{N}$ do
10:         $prompt = generate\_prompt\_2(K_i, Topic\_name\_list)$
11:         $K\_interact_i$ = LLM($prompt$)
12:     end for
13:     return $K\_interact$
14: end if
```

3 Experiments

In this section, we first introduce the experimental settings, including dataset, baselines, and implementation details, and then discuss the experimental results, as well as detailed analysis to demonstrate the effectiveness of LLM-DC.

3.1 Experimental Setup

Dataset. Our dataset originates from THUCNews [16], encompassing 840,000 news articles spread across 14 categories. We have chosen 12 out of these categories for our experiment, as done in previous work. Each category contains 2000 news which randomly sampled from each category of the original dataset. The 12 categories are sports, entertainment, lottery, real estate, education, fashion, politics, astrology, game, society, technology and stocks.

Table 1. Template of the interaction's prompt for categories correction. The different input for each category is highlighted in red, and the generated answers from LLM are displayed in blue.

System: Based on the provided keyword group, determine the document category that these keywords might indicate. These keywords are derived from a document collection that may contain different document categories. Our goal is to identify the document category names related to this batch of keywords. Please provide the category names directly without explaining the reasons.
Keywords: keywords_list

LLM: *Topic$_1$, …, Topic$_n$*

System: *Please filter out the key words related to Topic$_1$, …, Topic$_n$ from the following keywords.*
Keywords: keywords_list

LLM:
Topic$_1$: keywords_list$_1$
…
Topic$_n$: keywords_list$_n$

Table 2. Template of interaction's prompt for Keyword filtering. *Topic$_1$, ..., Topic$_m$* are confirmed category names. Different input keywords_list based on keywords extracted from each category. The LLM's generated answers in blue.

System: Please assign the keywords I provide to you into the categories of *Topic$_1$, …, Topic$_m$*
Keywords: keywords_list

LLM:
Topic$_1$: keywords_list$_1$
…
Topic$_n$: keywords_list$_n$

Baseline Methods. We evaluate LLM-DC against state-of-the-art models in the following two categories. **Self-learning based model (SelfL).** Self-learning can iteratively improve unsupervised document classification performance [17]. In this approach, documents are clustered using K-means, and keywords are extracted from them. Training data is then generated using keyword counting, followed by training a BERT-based classifier following a self-learning framework. This serves as the fundamental baseline for comparison. **Semi-supervised model (ClassKG).** ClassKG [14] is a framework for weakly-supervised text classification. It follows the similar way to ours for text classification: generating pseudo-labels, building a text classifier, and updating the keywords. ClassKG constructs a keyword graph to enhance pseudo-label generation and keyword updating by learning the correlations between keywords. It's worth noting that in both of these methods, the number of categories needs to be specified in advance.

Metrics. We used Precision (Pre), Recall(Rec) and F1 scores in evaluation. Precision can reflect the false alarm rate, while Recall can reflect the rate of missing report, and F1 is the average score of Precision and Recall.

Implementation Details. In LLM-DC we employ the DocSCAN for clustering. Using the GPT-3.5-Turbo API as the interface between our model and LLM. To identify the keywords for each category, we select the top 100 words with the highest TF-IDF scores. The threshold $diff_K$ in Eqn. 2 is set to be 0.2. In our document classification task, we employ the RoBERTa-wwm-ext [18] pre-trained language model as our classifier. The batch size for fine-tuning is set to 16. We employ the AdamW [19] optimizer, and the learning rate is 2e-6.

3.2 Experimental Results

We report the average performance of the proposed LLM-DC and other baseline models on THUCNews dataset. As we can observe from Table 3, LLM-DC shows stable and outstanding performance and achieves the state-of-the-art scores on most categories.

Table 3. Comparison of LLM-DC with baseline methods on THUCNews Dataset

Categories	SelfL			ClassKG			LLM-DC		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
sports	0.70	0.97	0.81	0.70	0.98	0.82	0.84	0.98	**0.90**
entertainment	0.90	0.94	0.92	0.90	0.94	0.92	0.91	0.95	**0.93**
lottery	0.97	0.60	0.74	0.97	0.60	0.74	0.98	0.83	**0.90**
real estate	0.89	0.94	0.91	0.92	0.93	**0.92**	0.91	0.93	**0.92**
education	0.88	0.96	0.92	0.90	0.95	**0.93**	0.87	0.96	0.92
fashion	0.95	0.92	**0.93**	0.92	0.93	0.92	0.95	0.91	**0.93**
politics	0.84	0.87	0.85	0.86	0.88	0.87	0.90	0.86	**0.88**
astrology	0.97	0.98	**0.98**	0.97	0.98	**0.98**	0.97	0.98	**0.98**
game	0.96	0.92	**0.94**	0.95	0.93	**0.94**	0.96	0.92	**0.94**
society	0.85	0.86	**0.85**	0.91	0.79	**0.85**	0.85	0.84	**0.85**
technology	0.89	0.65	0.75	0.84	0.82	**0.83**	0.87	0.70	**0.83**
stocks	0.83	0.91	0.87	0.87	0.89	0.88	0.85	0.92	**0.89**
Average	0.89	0.88	0.87	0.89	0.88	0.88	0.91	0.91	**0.90**

In the results of methods “SelfL” and “ClassKG”, we observed that their performance in classifying sports and lottery content was generally poor. Upon a deeper analysis of the dataset and its classification outcomes, we discovered that there is a certain degree of similarity between these two categories, such as sports lotteries being confused with sports content. Due to the lack of interaction and reliance solely on iterative processing, methods “SelfL” and “ClassKG” struggle to effectively distinguish categories that present these issues. In contrast, the proposed method, by introducing an interaction mechanism and leveraging the extensive pre-trained knowledge base of LLM, can substitute manual efforts to a certain extent, accurately differentiating between these similar yet subtly distinct keywords. The experimental results also confirm that our method achieved the highest F1 scores across almost all classification categories.

3.3 Effectiveness of LLM Feedback

Fig. 2 illustrates the changes in classification accuracy of LLM-DC throughout the entire interaction process. It can be observed that the initial classification accuracy is extremely low. That's because in the initial clustering stage, we didn't specify the number of categories, inevitably leading to noises from different classes being clustered into the same category. To address this issue, we implemented the LLM interaction strategy described in Section 2.4 during the first round of interaction, which corrected the category errors and significantly improved the classification accuracy. In subsequent interactions, we refined the selection of keywords extracted for each category based on the prompts in Table 2, which further enhanced the accuracy of the classifier, resulting in a 2% increase in precision. This experimental outcome validates the effectiveness of using LLM to optimize the number of categories and refine keyword selection.

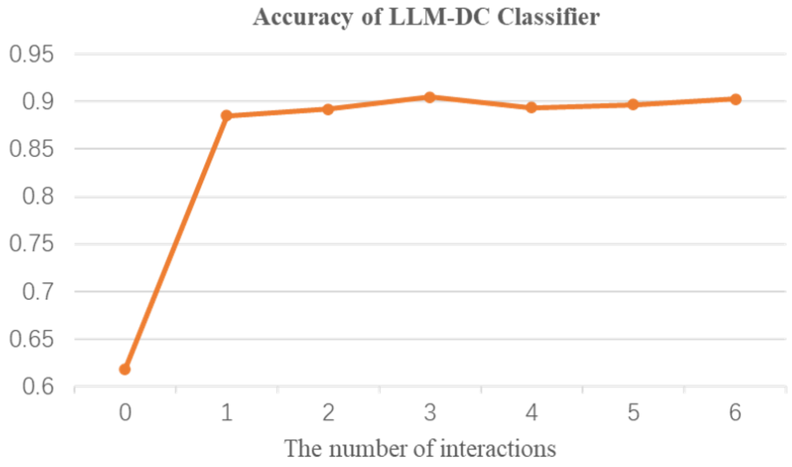

Fig. 2 Classification accuracy of LLM-DC throughout the entire interaction process.

To further demonstrate the effectiveness of LLM in correcting category errors, Table 4 provide the topics in clusters after clustering and the topics feedback by LLM. From the table, we can see that the initial clustering method mistakenly grouped documents from 12 categories into 10 categories. The “Cluster Topic” column indicates the distribution of ground truth labels within the cluster, while the “Topic Feedback by LLM” column displays the topic suggestions provided by LLM based on the prompts from Table 1. It was observed that LLM successfully separated the merged categories, offering topic suggestions that closely matched the actual topics. For example, within the topic suggestions for the “sports” category, the confusion between “lottery” and “sports” categories, as discussed in Section 3.2, was recognized during the initial interaction.

Table 4. Cluster topics and topic feedback by LLM.

Cluster Topics	Topic Feedback by LLM
Lottery, Game	Online Game, Lottery tickets
Real estate, Technology	Technology products, Real estate
Society	News report, Social events
Entertainment	Entertainment, File and television industry
Politics	Internet technology, International politics
Education	Education, Exam training
Fashion	Fashion, Clothing industry
Astrology	Astrology, Aquaculture
Stocks	Financial, Stock markets
Sports	Sports betting, football

4 Conclusion

In this paper, we propose an LLM-driven interactive document classification framework based on keyword feedback, which operates with minimal input—just the documents to be classified. We investigate the effectiveness of using LLM-generated feedback on interpretable keywords to enhance classification performance. The proposed interaction mechanism is seamlessly integrated into the document classification framework. Experimental results demonstrate its effectiveness in achieving accurate classification with LLM interaction. Our future research will focus on leveraging LLMs to identify and incorporate more effective keywords, thereby further enhancing classifier accuracy.

References

1. Li Q, Peng H, Li J, et al. A survey on text classification: From traditional to deep learning[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2022, 13(2): 1-41. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016)
2. Krishnan J, Purohit H, Rangwala H. Diversity-based generalization for neural unsupervised text classification under domain shift[C]//ECML-PKDD. 2020.
3. Schopf T, Braun D, Matthes F. Lbl2vec: An embedding-based approach for unsupervised document retrieval on predefined topics[J]. arXiv preprint arXiv:2210.06023, 2022.
4. Meng Y, Shen J, Zhang C, et al. Weakly-supervised neural text classification[C]//proceedings of the 27th ACM International Conference on information and knowledge management. 2018: 983-992.
5. Balcan M F, Blum A. Clustering with interactive feedback[C]//International Conference on Algorithmic Learning Theory. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008: 316-328.

6. Bouaziz A, da Costa Pereira C, Pallez C D, et al. Interactive generic learning method (IGLM) a new approach to interactive short text classification[C]//Proceedings of the 31st Annual ACM Symposium on Applied Computing. 2016: 847-852.
7. Zhang X, Li J, Chi P W, et al. ConceptEVA: Concept-based interactive exploration and customization of document summaries[C]//Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 2023: 1-16.
8. Mekala D, Shang J. Contextualized weak supervision for text classification[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 323-333.
9. Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. 2018.
10. Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.
11. Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.
12. Achiam J, Adler S, Agarwal S, et al. Gpt-4 technical report[J]. arXiv preprint arXiv:2303.08774, 2023.
13. Stammbach D, Ash E. Docscan: Unsupervised text classification via learning from neighbors[J]. arXiv preprint arXiv:2105.04024, 2021.
14. Zhang L, Ding J, Xu Y, et al. Weakly-supervised text classification based on keyword graph[J]. arXiv preprint arXiv:2110.02591, 2021.
15. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
16. Sun M, Li J, Guo Z, et al. Thuctc: an efficient chinese text classifier[J]. GitHub Repository, 2016.
17. Dong X L, De Melo G. A robust self-learning framework for cross-lingual text classification[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 6306-6310.
18. Xu Z. RoBERTa-WWM-EXT fine-tuning for Chinese text classification[J]. arXiv preprint arXiv:2103.00492, 2021.
19. Loshchilov I, Hutter F. Decoupled weight decay regularization[J]. arXiv preprint arXiv:1711.05101, 2017.